CLAIMS

1. A method of detecting overflow in a clamping circuit comprising the steps of:

inputting a first operand having a first fixed-point format into the clamping circuit;

inputting a second operand having a second fixed-point format into the clamping circuit;

determining an overflow output based upon the first and second fixed-point format and predicting whether an arithmetic operation of the first operand with the second operand will yield a result that exceeds the overflow output; and

performing at least partially the arithmetic operation of the first and second operands;

wherein the determining and predicting step occurs substantially in parallel with the performing step.

2. A method of detecting overflow in a clamping circuit comprising the steps of:

inputting a first operand having a first fixed-point format into the clamping circuit;

inputting a second operand having a second fixed-point format into the clamping circuit;

determining a product overflow output based upon the first and second fixed-point format and predicting whether multiplication of the first operand with the second operand yields a result that exceeds the product overflow output; and

performing at least partially the multiplication of the first and second operands;

wherein the determining and predicting step occurs substantially in parallel with the performing step.

3. A method of clamping fixed-point multipliers:

providing a first operand in a first fixed-point format;

providing a second operand in a second fixed-point format;

at least partially multiplying the first operand with the second operand to produce an operation result;

determining whether the operation result will exceed a representable value;

determining a clamping value based on the first fixed-point format of the first operand and the second fixed-point format of the second operand; and

substituting the operation result with the clamping value when it is determined that the operation result will exceed the representable value;

wherein the multiplying step and determining whether the operation result will exceed the representable value step occur substantially in parallel.

4. A method of clamping fixed-point multipliers:

providing a first and second input operand;

determining a desired number of output bits;

where any of the first and second input operands are positive, counting a number of leading logical zeros in the positive operand;

where any of the first and second input operands are negative, counting a number of leading logical ones in the negative operands;

summing the number of leading logical zeros of the positive input operands with the number of leading logical ones in the negative input operands;

determining a clamping decision based on the summing step to yield a simple clamp predictor representative of the clamping decision;

computing a product of the first operand and the second operand such that the product has the desired number of output bits plus one additional bit;

logically ORing the simple clamp predictor with a most significant bit of the product.

5. The method of claim 4 wherein the computing step and determining the clamping decision to yield the simple clamp predictor step occur substantially in parallel.

6. A method of processing multiplier data paths comprising the steps of:

performing at least a partial multiplication of a plurality of operands, each having a fixed-point format;

determining whether the at least partial multiplication of the operands produces a product that will exceed a pre-determined limit based upon the fixed-point format of each of the operands; and

wherein the performance step and the determining step occur substantially in parallel.

7. A method of clamp detection comprising the steps of:

inputting a first and a second operand to both a multiplier and an overflow detection circuit;

multiplying the first and second operands to generate a result not to exceed a pre-determined number of bits;

determining an initial clamping predictor bit based upon the first operand and the second operand; and

logically ORing the initial clamping predictor bit and a most significant bit of the result to produce a final clamping predictor bit;

wherein the multiplying and determining steps occur substantially in parallel.

8. The method of claim 7 wherein the first and second operands are in a fixed-point format.

9. The method of claim 7 wherein the most significant bit of the result is logically inverted prior to the logically ORing step.

10. The method of claim 8 wherein the step of determining the initial clamping predictor bit includes determining a number of logical zeros in each of the operands and summing the number of logical zeros to determine whether the sum exceeds a pre-determined number to determine the initial clamping predictor bit.

11. The method of claim 8 wherein the step of determining the initial clamping predictor bit includes determining a number of logical ones in each of the operands and summing the number of logical ones to determine whether the sum exceeds a pre-determined number.

12. The method of claim 8 wherein the step of determining the initial clamping predictor bit further comprises, when one of the operands is negative and one of the operands is positive, determining a number of logical ones for the negative operand and a number of logical zeros for the positive operand and summing the number of logical ones and the number of logical zeros to determine whether the sum exceeds or is equal to a pre-determined value for clamping to occur.

13. A multiplication overflow detection circuit comprising:

multiplication circuitry for at least partially multiplying a first and a second operand; and

overflow detection circuitry receiving the first and second operands that detects whether a result of the multiplication of the first and second operands exceed a maximum representable positive or negative value;

wherein the multiplication circuitry and the overflow detection circuitry operate substantially in parallel.

14. The circuit of claim 13 wherein the overflow detection circuitry utilizes a fixed-point format of the first and second operands to determine whether the result of the multiplication exceeds the maximum representable positive or negative value.

15. An overflow detection circuit comprising:

a first register for storing a first operand;

a second register for storing a second operand;

overflow detection circuitry for detecting an overflow of a multiplication of the first operand and the second operand and producing a clamp bit;

a multiplier for at least partially multiplying the first and second operands and generating a result not to exceed a pre-determined number of bits;

a clamp bit register for storing the clamp bit from the overflow detection circuitry; and

a result register connected to the multiplier for storing the result of the multiplication of the first and second operands;

wherein the overflow detection circuitry and the multiplier operate substantially in parallel.

16. The overflow detection circuit of claim 15 further comprising:

a multiplexer comprising:

a clamp value input for receiving a clamp value to be output when clamping occurs;

a clamp bit register input connected to the clamp bit register for receiving the clamp bit;

a result register input connected to the result register for receiving the result of the multiplication of the first and second operands; and

an output;

wherein the multiplexer selects one of the clamp value register input and the result register input based upon a logical level of the clamp bit register in order to make the selected input the output of the multiplexer.

17. The overflow detection circuit of claim 15 wherein the clamp bit input is logically ORed with a most significant bit of the result stored in the result register.

18. The overflow detection circuit of claim 15 wherein the registers are flip-flops.

19. The overflow detection circuit of claim 15 wherein the first and second registers store the first and second operands in a fixed-point format.

20. The method of claim 7 further comprising the step of determining whether clamping occurs based upon a logical value of the final clamping predictor bit and selecting one of a pre-selected clamp value and the result of the multiplying step.